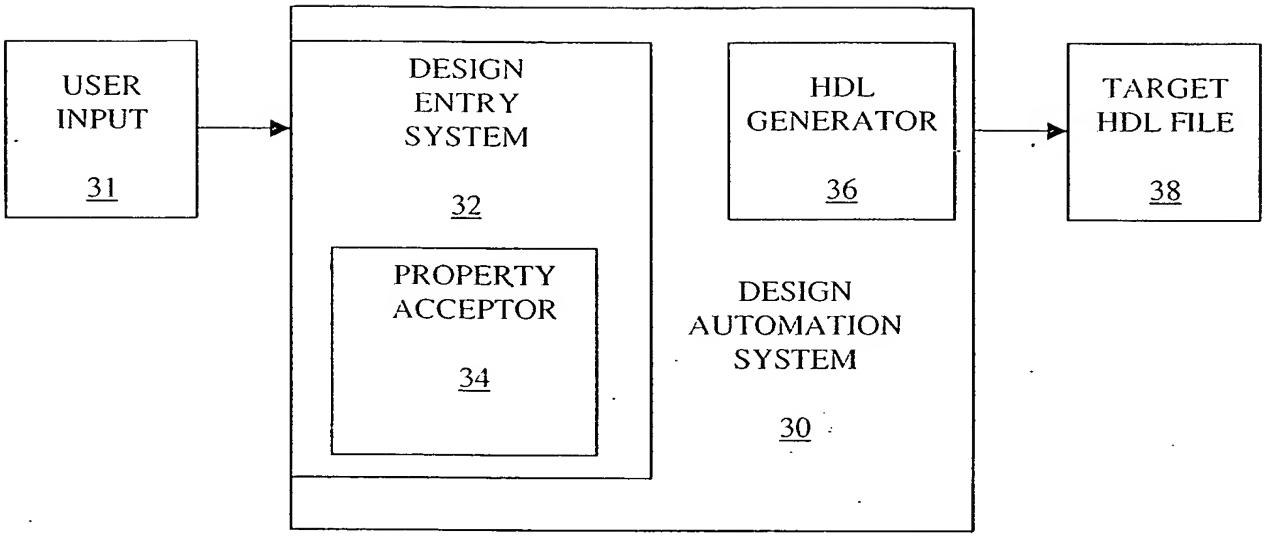
USER
INPUT
31
DESIGN
ENTRY
SYSTEM
32
PROPERTY
ACCEPTOR
34
HDL
GENERATOR
36
DESIGN
AUTOMATION
SYSTEM
30
TARGET
HDL FILE
38

Figure 1

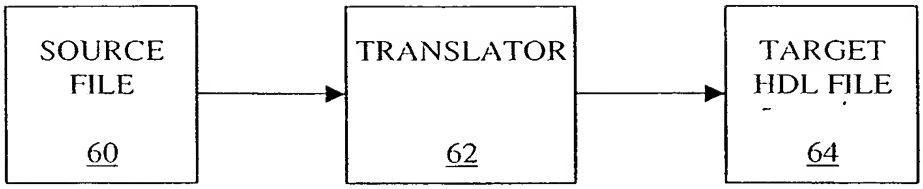

Figure 2

```
-------------------------------------------------
-- File Name : complex_counter_v1.vhd
-- Update to : 21-Jan-2004
-- Designer  : Yinon Levy @ Royal Design
-------------------------------------------------

 library ieee ;
 use ieee.std_logic_1164.all;
 use ieee.std_logic_unsigned.all;
 use ieee.std_logic_arith.all;
 use work.Rx_package.all;

10

 entity complex_counter_v1 is port

 (
  ck          : in  std_logic                      ; -- System clock
  mrstn       : in  std_logic                      ; -- Asynchronous reset
  en1         : in  std_logic                      ; -- Enable
  count       : out std_logic_vector(5 downto 0)   -- Counter value
 );
 end complex_counter_v1 ;

 architecture hdm of complex_counter_v1 is

 signal p_count :std_logic_vector(5 downto 0) ;

 begin

 count <= p_count ;

 -------------------------------------------------
 rx_complex_counter: process (ck, mrstn)

12

 procedure reset_procedure  is
 begin
14     p_count  <= (others=> '0')  ;
 end reset_procedure             ;

 begin

16

  RxProperties (ClkName=>"ck", ClkPolarity=> High,
18              RstName=> "mrstn", RstPolarity=> Low,
                SyncMode=>False,RstProc=>"reset_procedure");

  ----------- Rx Statement -----------------------------------------
  RxStatement   (StatementName=> "count_statement", HitEventMode=> Serial) ;

20

  HitEvent        (StatementName=>"count_statement",
                   HitEventName=>"count_hit_event") ;

22

  Trigger         (HitEventName=>"count_hit_event",
                   TriggerExp=>%  ##1 en1='1' ; ## 2 % );

24

  RollingAction   (HitEventName=>"count_hit_event",
                   TimeValueAction=> % ##12 p_count <= p_count + 1 %);

26

 end process rx_complex_counter ;
 end hdm ;
```

Figure 3

```
-------------------------------------------------
-- File Name : complex_counter_v1.vhd
-- Update to : 21-Jan-2004
-- Designer  : Yinon Levy @ Royal Design
-------------------------------------------------

 library ieee ;
 use ieee.std_logic_1164.all;
 use ieee.std_logic_unsigned.all;
 use ieee.std_logic_arith.all;
--use work.Rx_package.all;

40 entity complex_counter_v1 is port

 (
  ck          : in  std_logic                      ; -- System clock
  mrstn       : in  std_logic                      ; -- Asynchronous reset
  en1         : in  std_logic                      ; -- Enable
  count       : out std_logic_vector(5 downto 0)   -- Counter value
 );
 end complex_counter_v1 ;

 architecture hdm of complex_counter_v1 is

 signal p_count :std_logic_vector(5 downto 0) ;

 begin

 count <= p_count ;

-------------------------------------------------
P_rx_complex_counter: process (ck, mrstn)


 procedure reset_procedure  is
 begin
    p_count  <= (others=> '0')  ;
 end reset_procedure            ;

   -- Handshake Rx-defined signals
   variable    rd_counter_count_hit_event_0       : unsigned(3 downto 0) ;
   variable    rd_end_trig_count_statement        : std_logic ;
   variable    rd_end_exec_count_statement        : std_logic ;


   -- ---------------------------------------------------------------------
   --  Rx Reset Procedures
   -- ---------------------------------------------------------------------
   procedure rd_rst_count_statement  is
   begin
      rd_end_trig_count_statement := '0';
      rd_end_exec_count_statement := '0';
      rd_counter_count_hit_event_0 := (others=> '0');
   end rd_rst_count_statement;

   -- ---------------------------------------------------------------------
   -- Statement Name: count_statement
   -- Statement Mode: Parallel
   -- ---------------------------------------------------------------------
```

Figure 4a

```
-- ------------------------------------------------------------------------
-- Hit Event Name:      count_hit_event
-- Trigger Expression: ##1  en1='1'   ## 2
-- ------------------------------------------------------------------------

procedure count_hit_event_evaluate  is
begin
   if (rd_counter_count_hit_event_0=0) then
      if (en1='1') then
         rd_counter_count_hit_event_0 := rd_counter_count_hit_event_0 + 1;
      end if;
   elsif (rd_counter_count_hit_event_0<=2) then
      rd_counter_count_hit_event_0 := rd_counter_count_hit_event_0 + 1;

      if (rd_counter_count_hit_event_0=3) then
         rd_counter_count_hit_event_0 := (others=> '0');
         rd_end_trig_count_statement := '1';
      end if;
   end if;
end count_hit_event_evaluate;

procedure count_hit_event_execute  is
begin
   if (rd_counter_count_hit_event_0<=11) then
      rd_counter_count_hit_event_0 := rd_counter_count_hit_event_0 + 1;
      p_count <= p_count+1;
   end if;
   if (rd_counter_count_hit_event_0=12) then
      rd_end_exec_count_statement := '1';
   end if;
end count_hit_event_execute;

procedure count_hit_event_activate  is
begin
   if (rd_end_trig_count_statement = '0') then
      count_hit_event_evaluate;
   end if;
   if (rd_end_trig_count_statement = '1'
       and rd_end_exec_count_statement = '0') then
      count_hit_event_execute;
   end if;
   if (rd_end_exec_count_statement = '1') then
      rd_rst_count_statement;
   end if;
end count_hit_event_activate;

-- ------------------------------------------------------------------------
-- Rx Statements Activation Process
-- ------------------------------------------------------------------------
begin
   if (mrstn = '0') then
      reset_procedure;
      rd_rst_count_statement;
   elsif (rising_edge(ck)) then
      count_hit_event_activate;
   end if;
end process;
end hdm ;
```

102

104

106

108

Figure 4b

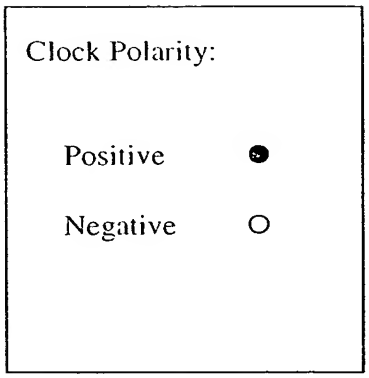

Figure 5

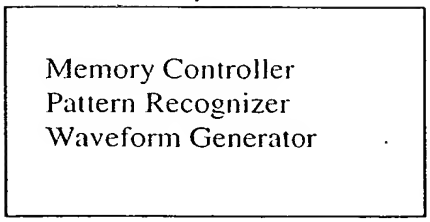

Figure 6

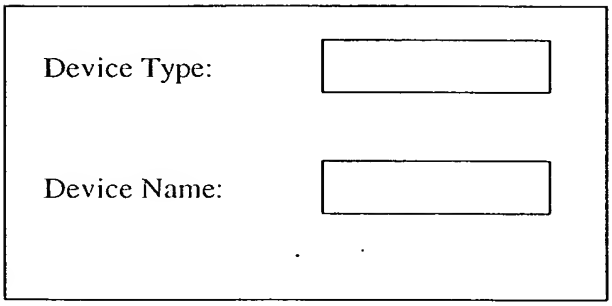

Figure 7

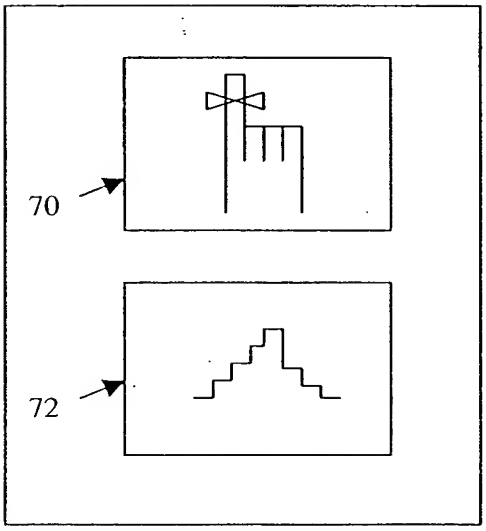

Figure 8

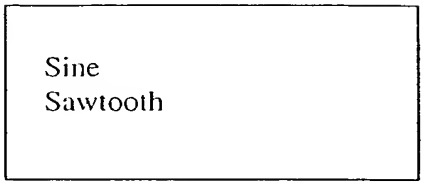

Figure 9

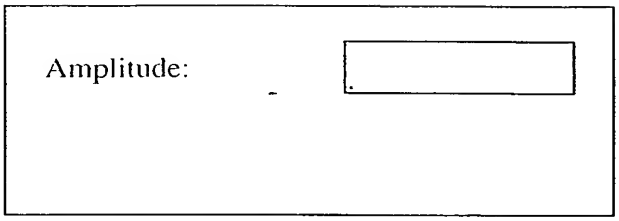

Figure 10

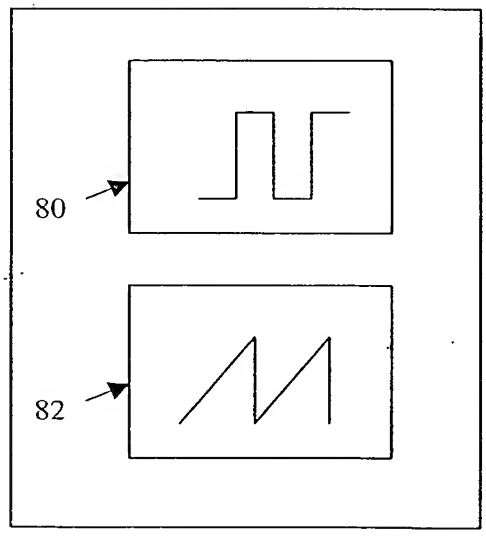

Figure 11

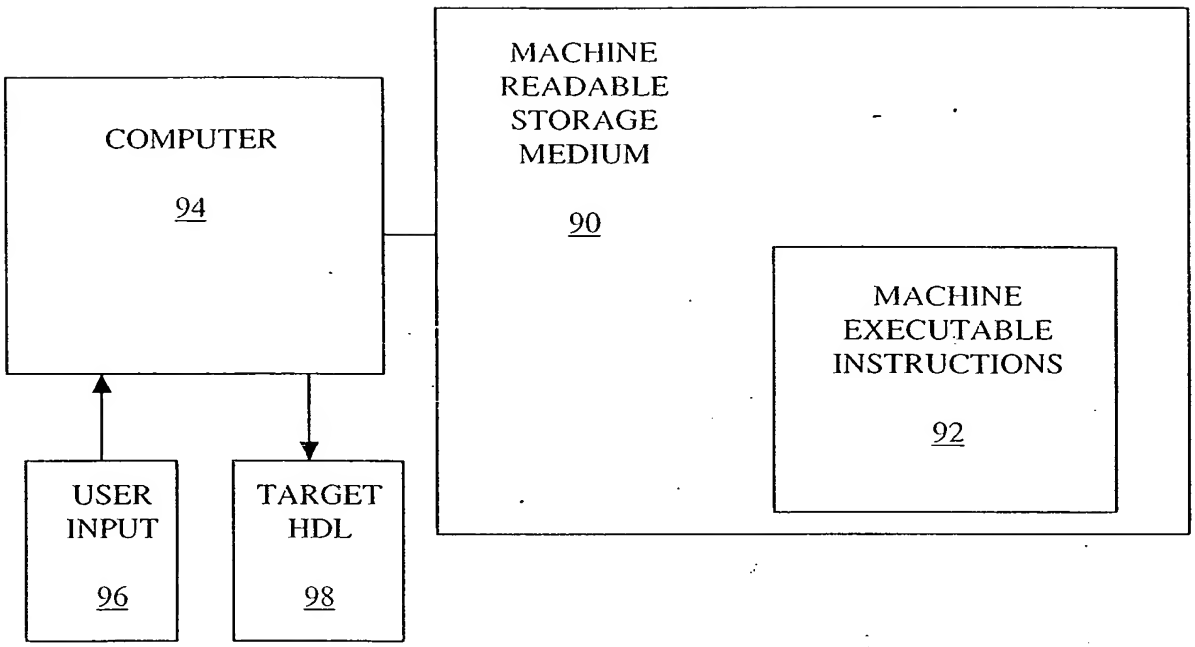
COMPUTER
94
MACHINE
READABLE
STORAGE
MEDIUM
90
MACHINE
EXECUTABLE
INSTRUCTIONS
92
USER
INPUT
96
TARGET
HDL
98

Figure 12